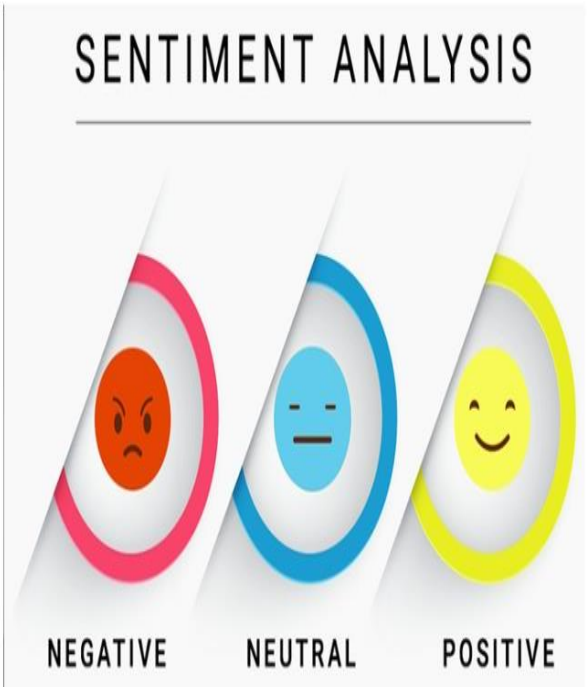
PHASE 4 PROJECT SUBMISSION

PROJECT TITLE : SENTIMENT ANALYSIS FOR MARKETING

PHASE 4: DEVELOPMENT PART-2

TOPIC: Continue building the sentiment analysis for marketing model using python by feature engineering,model training and evaluation.



INTRODUCTION:

Sentiment analysis is a marketing tool that helps you examine the way people interact with a brand online. This method is more comprehensive than traditional online marketing tracking, which measures the number of online interactions that customers have with a brand, like comments and shares. Using sentiment analysis, you can label individual interactions as positive, negative or

neutral. Once you've figured out how to determine and track these labels, you can use this new data set for a variety of marketing purposes, including your online strategy. Sentimental analysis is an extremely useful tool to have since higher numbers of interactions don't always equate to better results. For example, if you were to receive 10 replies on a social post and all of them were positive, your post likely had a more compelling effect on your audience than if you receive 100 replies with only 10 of them being positive. The primary purpose of sentiment analysis is to respond to commentary more constructively.

3 types of sentiment analysis

To perform sentiment analysis, a marketing team might use a software platform that creates an algorithm to monitor customer engagement online. There are three fundamental ways to develop an algorithm for distinguishing social sentiment:

- **Manual analysis:** This type uses manually created rules based on neurolinguistic principles, such as stemming and tokenization. It takes a long time to set up, but it's easy to change and customize.

- **Automatic analysis:** This type uses machine learning techniques that use neural networks and statistical models to classify language. It can be challenging to change, but it's easy to set up and manage.

- **Hybrid analysis:** This type uses both rules-based and machine-learning analyses. It's a balanced approach that most social listening applications employ.

FEATURE ENGINEERING: Feature engineering is crucial for sentiment analysis in marketing. Here are some key features to consider:

1. Text Preprocessing:

- ❖ Tokenization: Split text into words or subword tokens.
- ❖ Stopword Removal: Eliminate common words that don't carry sentiment.
- ❖ Lemmatization or Stemming: Reduce words to their base form.

2. N-grams:

- Unigrams, bigrams, and trigrams can capture word combinations and context.

3. Bag of Words (BoW) or TF-IDF:

- BoW represents word counts in a document, while TF-IDF considers term frequency and inverse document frequency to weigh word importance.

4. Word Embeddings:

- Word2Vec, GloVe, or fastText embeddings can capture semantic relationships between words.

5. Sentiment Lexicons:

- Use sentiment dictionaries like AFINN, SentiWordNet, or the VADER lexicon to assign sentiment scores to words.

6. Part-of-Speech (POS) Features:

- Analyze the distribution of nouns, verbs, adjectives, and adverbs in the text.

7. Sentiment Polarity:

- Calculate the overall sentiment polarity of the text using methods like TextBlob or VADER.

8. Emoticons and Emoji Analysis:

- Extract and analyze emoticons and emojis as they can carry sentiment.

9. Capitalization and Punctuation:

- Consider features related to capitalization and the use of punctuation.

10. Text Length:

- The length of the text can sometimes correlate with sentiment.

11. Brand/Product Mentions:

- Identify mentions of specific brands or products within the text.

12. Topic Modeling:

- Discover topics within the text that might affect sentiment.

13. Social Media Features:

- For social media sentiment analysis, features like retweets, likes, and comments can be informative.

14. User Information:

- If available, user demographics or profiles can add context to sentiment analysis.

15. Time and Trends:

- Analyze sentiment trends over time to identify patterns and seasonality.

16. Competitor Mentions:

- Monitor mentions of competitors to gauge sentiment in a competitive context.

17. Contextual Features:

- Incorporate context, such as the source of the text (e.g., reviews, tweets, comments), to better understand sentiment.

18. Domain-Specific Features:

- Consider industry-specific terminology and jargon that may impact sentiment.

Remember to adapt your feature engineering to the specific needs of your marketing sentiment analysis task, whether it's for social media monitoring, customer reviews, or other sources of textual data. Machine learning models can then be trained using these engineered features to predict sentiment effectively.

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split

# Load your Kaggle dataset
data = pd.read_csv("your_kaggle_dataset.csv")

# Feature engineering
# Example: Text preprocessing and feature extraction
```

```
data['text'] = data['text'].apply(lambda x: x.lower())  # Convert text to lowercase
data['word_count'] = data['text'].apply(lambda x: len(x.split()))  # Word count feature

# Split the data into training and testing sets
X = data['text']
y = data['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Vectorize the text data
vectorizer = TfidfVectorizer(max_features=1000, stop_words='english')
X_train_tfidf =vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Now, X_train_tfidf and X_test_tfidf contain the TF-IDF features

# Train your sentiment analysis model (e.g., using scikit-learn or other libraries)
```

Once the model is trained, you can use it to make predictions on new data.

For example:

predictions = your_model.predict(X_test_tfidf)

Evaluate the model and continue with further feature engineering as needed.

MODULE TRAINING:

To train a sentiment analysis model for marketing, you can follow these steps:

1. Data Preparation:

- Collect and preprocess marketing-related text data, such as customer reviews, social media comments, or surveys.

- Label the data with sentiment scores (e.g., positive, negative, neutral).

2. Text Preprocessing:

- Tokenize the text data.

- Remove stopwords and special characters.

- Apply stemming or lemmatization.

- Convert text to numerical format using techniques like TF-IDF or word embeddings.

3. Feature Engineering:

- Create relevant features (as mentioned in the previous response) based on your dataset's characteristics.

- Generate features that capture the context and nuances of marketing sentiment.

4. Split Data: - Divide the dataset into training, validation, and test sets.

5. Select a Model:

- Choose a machine learning or deep learning model suitable for sentiment analysis, such as logistic regression, Naive Bayes, support vector machines, recurrent neural networks (RNNs), or transformer-based models (e.g., BERT, GPT-3).

6. Model Training:

- Train your chosen model on the training dataset.

- Fine-tune hyperparameters as needed to optimize performance.

- Use cross-validation to assess model performance during training.

7. Model Evaluation:

- Evaluate the model on the validation dataset to measure accuracy, precision, recall, F1-score, and other relevant metrics.

- Adjust the model based on the validation results.

8. Hyperparameter Tuning:

- Experiment with different hyperparameters to optimize the model's performance.

9. Test the Model:

- Finally, evaluate the model on the test dataset to assess its real-world performance.

10. Deployment:

- Once you are satisfied with the model's performance, deploy it for real-time sentiment analysis in your marketing context. This can be integrated into marketing tools or platforms for automated analysis.

11. Monitoring and Iteration:

- Continuously monitor the model's performance and retrain it periodically with new data to adapt to changing sentiments in marketing.

Remember that the choice of model and feature engineering techniques may vary depending on the size of your dataset, the specific marketing data sources, and the level of accuracy and interpretability required. Additionally, you can make use of pre-trained models for sentiment analysis to jumpstart your project, especially if you have limited data.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

# Load your preprocessed dataset
data = pd.read_csv("your_preprocessed_data.csv")

# Split the data into training and testing sets
X = data['text']
y = data['sentiment']  # Replace 'sentiment' with the actual sentiment label column name
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Vectorize the text data
vectorizer = TfidfVectorizer(max_features=1000, stop_words='english')
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_testX_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

```
# Train a sentiment analysis model (Logistic Regression in this example)
model = LogisticRegression()
model.fit(X_train_tfidf, y_train)

# Make predictions on the test data
predictions = model.predict(X_test_tfidf)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
report = classification_report(y_test, predictions)

print(f"Accuracy: {accuracy}")
print(f"Classification Report:\n{report
```

EVALUATION OF SENTIMENTAL ANALYSIS:

Evaluating a sentiment analysis model for marketing is crucial to assess its performance and determine its effectiveness in capturing consumer sentiment. Here are some key evaluation metrics and steps for model evaluation in a marketing context:

1. Data Splitting:

- Split your dataset into training, validation, and test sets. Common splits are 70% for training, 15% for validation, and 15% for testing.

2. Evaluation Metrics:

- Choose appropriate evaluation metrics, depending on your sentiment analysis task. Common metrics include:

- Accuracy: The proportion of correctly classified samples.

- Precision: The number of true positives divided by the total number of predicted positives.

- Recall: The number of true positives divided by the total number of actual positives.

- F1-Score: The harmonic mean of precision and recall.

- Confusion Matrix: To analyze false positives, false negatives, true positives, and true negatives.

3. Model Evaluation:

- Use the test dataset to evaluate your sentiment analysis model. Calculate the selected evaluation metrics to assess its performance.

- Analyze the confusion matrix to understand the types of errors made by the model.

- Consider ROC curves and AUC scores for binary sentiment analysis.

4. Cross-Validation:

- Perform k-fold cross-validation on the training data to get a more robust assessment of your model's performance. This helps ensure your model generalizes well to unseen data.

5. Domain-Specific Metrics:

- In marketing, consider domain-specific metrics such as Net Promoter Score (NPS), Customer Satisfaction Score (CSAT), or customer retention rates for understanding the impact of sentiment on business outcomes.

6. A/B Testing:

- In marketing, you can conduct A/B testing to measure the real-world impact of your sentiment analysis model on marketing campaigns or strategies.

7. Interpretable Models:

- If model interpretability is important, consider using interpretable models like decision trees or logistic regression to understand how the model makes predictions.

8. Fine-Tuning and Hyperparameter Tuning:

- Adjust model hyperparameters and perform hyperparameter tuning to improve performance.

9. Handling Class Imbalance:

- Address class imbalance issues, especially if one sentiment class is significantly smaller than the others. You may need to oversample, undersample, or use techniques like SMOTE.

10. Error Analysis:

- Analyze specific examples where the model made mistakes to gain insights into areas for improvement.

11. Model Deployment and Monitoring:

- After achieving satisfactory results, deploy your model in a real marketing context. Monitor its performance over time and update as necessary.

12. Continuous Improvement:

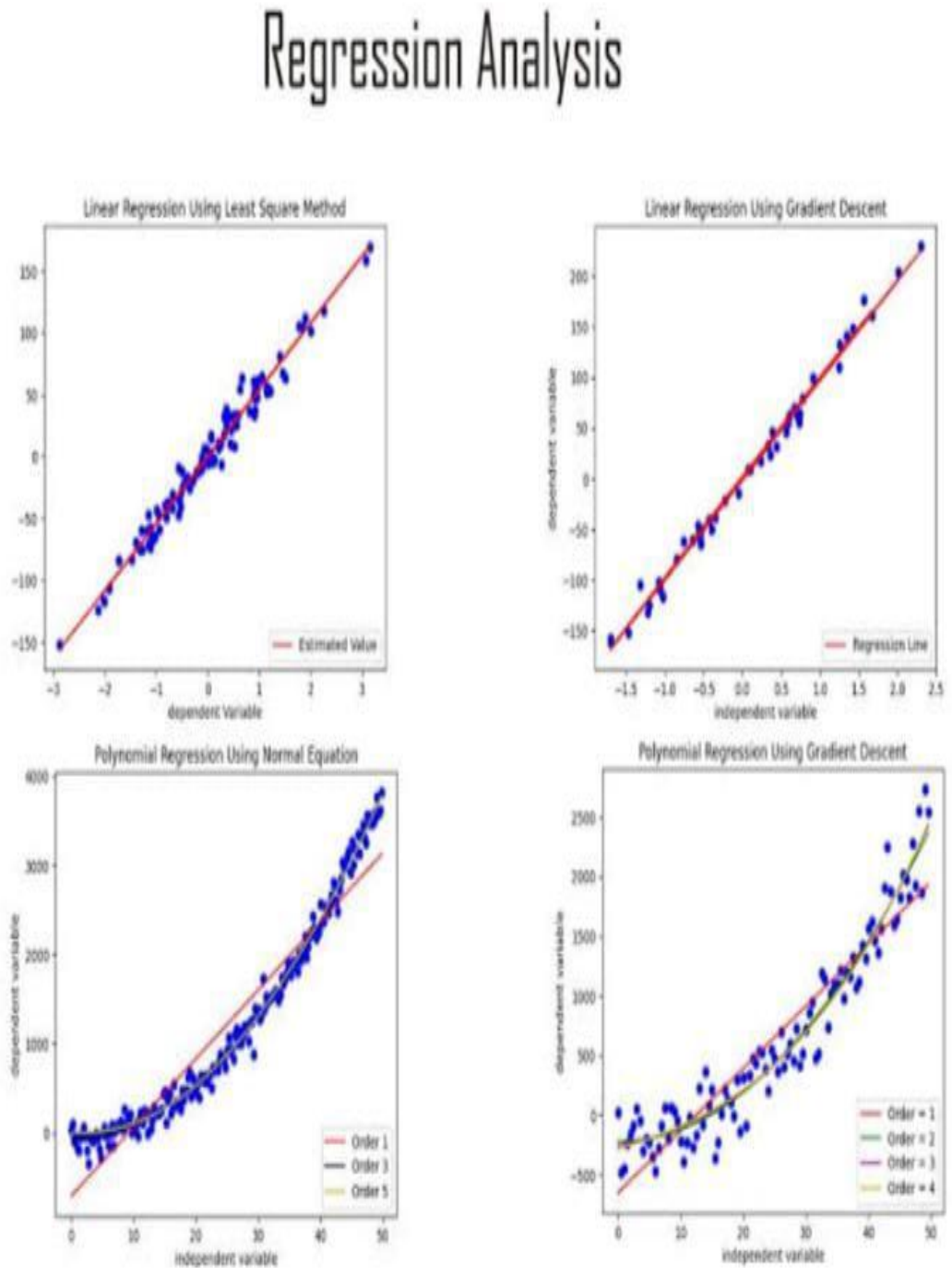
- Continuously collect new data and retrain your model to adapt to changing consumer sentiment.

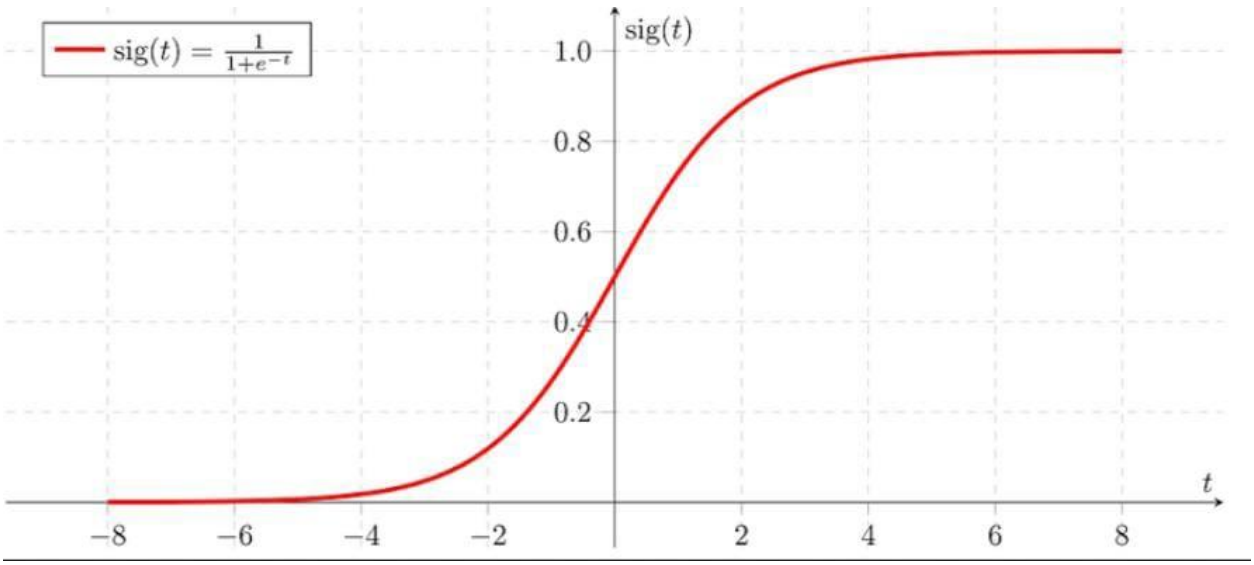
By following these steps and using appropriate evaluation metrics, you can effectively assess the performance of your sentiment analysis model in a marketing context and make informed decisions to improve your marketing strategies.

```
import pandas as pd
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.externals import joblib  # For model loading

# Load your test data and model
```

```
test_data = pd.read_csv("test_data.csv")  # Replace with your test dataset

model = joblib.load("sentiment_model.pkl")  # Replace with your trained model


# Assuming your test data has columns 'text' and 'sentiment'

X_test = test_data['text']

y_true = test_data['sentiment']


# Preprocess the test data (similar to how you preprocessed the training data)

# For example, tokenize, remove stopwords, and vectorize using the same vectorizer


# Make predictions using the pre-trained model

predictions = model.predict(X_test)


# Evaluate the model

accuracy = accuracy_score(y_true, predictions)

report = classification_report(y_true, predictions)

confusion = confusion_matrix(y_true, predictions)


print(f"Accuracy: {accuracy}")
```

```
print(f"Classification Report:\n{report}")
print(f"Confusion Matrix:\n{confusion}")
```





CONCLUCION:

Collecting large amounts of unstructured data from various sources. Tracking real-time customer feedback and sentiment about an organization's brand, products and services. Providing feedback on ways to improve products, services and customer experience.